

CLAIMS:

1. A method for operating three-dimensional (3D) application software intended to provide a display output to a two-dimensional (2D) screen display comprising:

(a) running the application software in its normal mode to generate 3D application data output which is normally to be sent to an application programming interface (API) driver for the 2D screen display;

(b) intercepting the 3D application data output from the application software and redirecting the data to a pseudo driver for generating a 3D stereoscopic display; and

(c) using the pseudo 3D display driver to generate a 3D stereoscopic display.

2. A method according to Claim 1, wherein the 3D stereoscopic display is selected from the group consisting of head-mounted "stereo vision" goggles, head-mounted 3D display device, and a stereo vision monitor.

3. A method according to Claim 1, wherein the 3D application software is a 3D video game software which provides 3D game data output.

4. A method according to Claim 3, wherein the intercepting and redirecting of the 3D game data is obtained by providing a wrapper for the game software's native API having stereoscopic display function calls linked under the same name as the game software's native API for 2D display.

5. A method according to Claim 4, wherein the wrapper supports a selected one of the following group of native API formats: Glide; OpenGL; and DirectX.

6. A method according to Claim 1, wherein the pseudo driver generates a 3D stereoscopic display using separate graphics cards for rendering right and left image viewpoints for the 3D stereoscopic display.

5 7. A method according to Claim 1, wherein the pseudo driver generates a 3D stereoscopic display using one graphics card with dual heads for rendering right and left image viewpoints for the 3D stereoscopic display.

10 8. A method according to Claim 3, wherein the intercepted 3D game data is stored in a 3D data recorder for later play back.

9. A method according to Claim 8, wherein the recorded 3D game data are transmitted or downloaded through an online interface to a remote user.

15 10. A method according to Claim 3, wherein the intercepted 3D game data is combined with other 3D content using a mixer and a dual rendering system.

20 11. A method according to Claim 10, wherein the dual rendering system is kept running while switching between different game software.

12. A method according to Claim 3, wherein another pseudo driver operates on the 3D game data in tandem with the pseudo 3D display driver.

25 13. A method according to Claim 12, wherein the other pseudo driver is a stereo sound or a directional force feedback driver.

14. A method according to Claim 12, wherein the video game software is run with one or more tracking devices for input from the player.

15. A 3D display system for operating three-dimensional (3D) application software which makes display function calls to a native API for the software under an API linking name to provide a display output to a two-dimensional (2D) screen display comprising:

(a) a computer for running the application software in its normal mode to generate
5 3D application data output;

(b) a file directory system for the computer in which the application software's native API is normally stored under the API linking name; and

(c) a pseudo 3D display driver stored in the computer's file directory system under the API linking name as a wrapper in place of the native API for intercepting the display function
10 calls and 3D application data output from the application software and redirecting them through the pseudo 3D display driver in order to generate a 3D stereoscopic display.

16. A 3D display system according to Claim 15, wherein the 3D stereoscopic display is selected from the group consisting of head-mounted "stereo vision" goggles, head-
15 mounted 3D display device, and a stereo vision monitor.

17. A 3D display system according to Claim 15, wherein the 3D application software is a 3D video game software which provides 3D game data output.

18. A 3D display system according to Claim 17, wherein the wrapper supports a selected one of the following group of native API formats: Glide; OpenGL; and DirectX.
20

19. A 3D display system according to Claim 15, wherein the pseudo 3D display driver specifies right and left eye views for the 3D application data output, and sets up parallel
25 rendering engines using the native API for converting the right and left eye views into right and left image data, respectively, which are used for the 3D stereoscopic display.

20. A 3D display system according to Claim 19, further including separate graphics cards for rendering right and left image displays for the 3D stereoscopic display.